Volume Rendering: the added value of Stereo in real-life Applications

Koen Poppe   *Gert Van Hoey (Barco)*

Report TW 563, April 2010

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Volume Rendering: the added value of Stereo in real-life Applications

Koen Poppe　　*Gert Van Hoey (Barco)*

Report TW 563, April 2010

Department of Computer Science, K.U.Leuven

Abstract

Volume rendering is a well known visualization technique for volumetric scalar data. Several publications already described the acceleration of this technique when using the graphical processing unit (GPU). This enables real-time viewpoint manipulations which, in turn, facilitates the interpretation.

Stereo, where two slightly different images are generated for each eye, proved to be a valuable addition to this. Mostly because, apart from shading and motion parallax, stereo is one of the most important depth cues available in this context.

We have written a fully object-oriented C++ framework that contains software for both GPU accelerated volume rendering and stereo image generation. Using this, the benefits of stereo are presented in a qualitative fashion. One example show how it helps doctors with the examination of CT scans, a second illustrates the benefits for other applications, here the analysis of volume data originating from material science simulations.

Keywords : Volume rendering, GPU, OpenGL Shading Language (GLSL), Object oriented C++ Framework, CT-scan visualisation, grain growth visualisation.
MSC : Primary : 65D18

Volume Rendering: the added value of Stereo in real-life Applications

Koen Poppe*, Gert Van Hoey§

***Abstract*—Volume rendering is a well known visualization technique for volumetric scalar data. Several publications already described the acceleration of this technique when using the graphical processing unit (GPU). This enables real-time viewpoint manipulations which, in turn, facilitates the interpretation.**

Stereo, where two slightly different images are generated for each eye, proved to be a valuable addition to this. Mostly because, apart from shading and motion parallax, stereo is one of the most important depth cues available in this context.

We have written a fully object-oriented C++ framework that contains software for both GPU accelerated volume rendering and stereo image generation. Using this, the benefits of stereo are presented in a qualitative fashion. One example show how it helps doctors with the examination of CT scans, a second illustrates the benefits for other applications, here the analysis of volume data originating from material science simulations.

I. Volume Rendering

Using the graphical processing unit (GPU) for volume rendering is not new: the earliest publications date back to 1993 ([5] gives a good historical overview). It is still an active topic: recent developments include shadowing [4], framework implementations [7], frequency domain rendering [9], ...

Without going into technicalities (see [2] for more details), volume rendering can be understood as follows. For every pixel in the image, a ray is created from the camera through that pixel. Following this ray, the volumetric data is sampled using a three-dimensional texture which contains the data. Because texture memory operations are highly optimized on graphics cards, this sampling is very efficient. The resulting samples are then transformed in a certain fashion to produce the color for the pixel, as shown schematically in Figure 1. Some of the best known transformations of the samples are

- *Maximum Intensity Projection (MIP)*, where only the largest sample on the ray determines the final color,
- *Depth Weighted MIP*, where that largest sample is first scaled based on its distance to the camera, *and*
- *Compositing*, where each sample is transformed to a (mostly semi-transparent) color that is accumulated over the ray to form the final color for that pixel.

Implementing an algorithm on a GPU has always been rather awkward: vendor specific extensions had to be used and portability could therefore not be guarantied. The introduction of the OpenGL Shading Language (GLSL) solved this by providing a consistent API for developers. Nearly all

* K. Poppe is with the department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium. E-mail: Koen.Poppe@cs.kuleuven.be .

§ G. Van Hoey works at the Barco Technology Center, President Kennedypark 35, 8500 Kortrijk, Belgium. E-mail: Gert.VanHoey@barco.com .

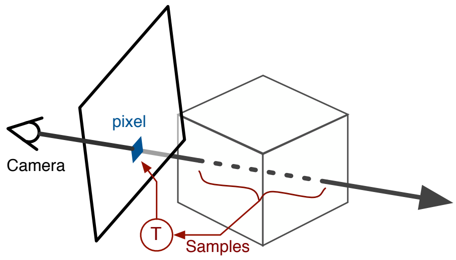


Figure 1. The basics of volume rendering: samples along a ray from the camera through each pixel are traced and combined to form that pixels color.

recent graphics cards support this and by focussing on these specifications, the fully object-oriented C++ software for this research is made portable. It is successfully tested by several persons, on different machines, a variety of graphics cards and running on the three most commonly used operating systems.

II. Adding Stereo

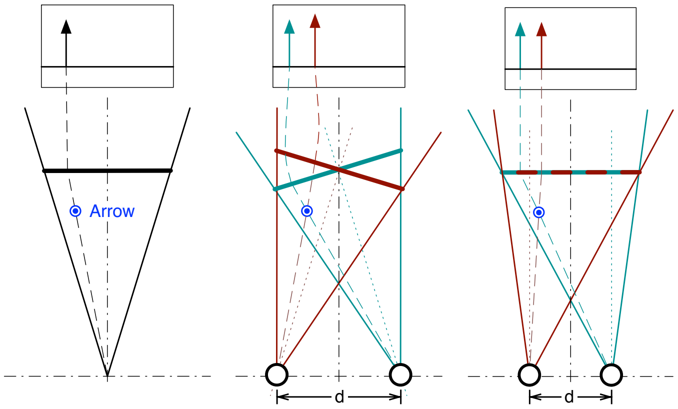
Besides motion parallax (accomplished either by head tracking or by manipulation of the viewpoint) stereo is an important depth cue. Research with three dimensional nets [10] showed that, with the addition of these cues, the interpretation improved by respectively 120% and 60% over (mono) still images. Moreover, combining both cues led to an improvement of 200%. This coincides with our observation that the combination of the cues leads to a far better understanding.

Several kinds of stereo images can be generated. They differ in the positioning of the virtual cameras for each eye and the way the correct image is transferred to the corresponding eye.

A. Projection

The easiest way to generate stereo is the so called *Toe-in*-method: it uses two camera's that look at the same point but are separated by a distance proportional to the eye separation (see Figure 2(b)). This has some disadvantages: as can be seen from the figure, the projection planes do not correspond. This leads to a vertical parallax: the projection for the right eye will be smaller than in the image for the left one when considering the upwards pointing arrow located in the scene. This effect is worst in the corners and makes it uncomfortable to use.

An alternative, as shown in Figure 2(c), uses parallel axes and an asymmetric frustum. As can be seen from the figure, vertical parallax is not an issue here. This projection method results in more relaxed images and is therefore preferred.



(a) Mono. (b) Toe-in. (c) Asymmetric frustum, parallel axes.

Figure 2. Mono -and two stereo projections of the upwards pointing arrow. Note the vertical parallax for the *Toe-in* stereo projection: the length of the projected arrow differs for the left and right view. This is not the case in (c).

B. Encoding

Now both views must be addressed to the corresponding eye. Again, several alternatives exist. The oldest one used *time-multiplexing*: the monitor alternates between the left and right image. Using synchronized shutter glasses, the eye that should not see the current image will be covered. Refresh rates of at least 120Hz are required for stable image. Liquid Crystal Displays (LCD) are not very well suited for this due to their refreshing paradigm (subsequent images are interleaved).

A second approach encodes the images in horizontal and vertical polarized light. A possible setup for this consists of two regular LCD's with a polarizer and a semi-transparent mirror. The images are split using glasses with corresponding analyzers. A similar setup is used during this research.

A third approach is widely known: *anaglyph* or color encoding transfers both images in a different coloring (typically red en cyan, but other color schema's exist). Using filter glasses with the same colors, the images for both eyes can be separated. Note that color information from the image will interfere with the encoding: not every color can be sent to each eye. It also takes some time to get used to differences in color for each eye and its relative luminance. We refer to [1] and [11] for more details on color encodings and related problems.

C. Parameters

Several parameters determine the creation of stereo images:

- *Eye separation* – The distance between the virtual camera's affects the stereo: if d (see Figure 2) is small, no stereo is generated, if d is too large, the human brain will fail to combine both images. The optimum depends on the users viewing distance and the size of the screen.
- *Position of projection plane* – If the projection plane is placed behind the object, everything will appear to *come out* of the screen; if it is chosen before, all will be located *in* the screen. The best position, as found in [12], is so that one third of the object is before the projection plane. This can be related to the *depth-of-field* effect in photography.
- *Depth compression* – When the stereo effect is too aggressive (especially when visualizing long object along their longest axes) one can apply a non-uniform scaling in the viewing direction. This compresses the apparent depth of the object and facilitates the interpretation of it.

A final paragraph on stereo should be addressed to the *accomodation-convergence mismatch*. This is the main reason why users need time to get used to stereo images. In real-life situations, the accommodation (the focussing of the eye's lens) is linked to the convergence (the inwards rotation of the eyes) because the distance to the subject determines both. For stereo images, this is no longer true: the convergence is adapted to the position of the object relative to the projection plane, but the accommodation does not change as the screen itself remains at a fixed distance. A detailed description can be found in [3].

III. Applications

A. Medical imaging

Imaging techniques are widely used recently. Because of their non-invasiveness, they provide much data with only minimal side effects for the patient. However, for specialists, the data gathered is rarely straightforward to analyze. First of all, the size of the scan data can be a problem (1GB per scan is not uncommon). It challenges manufacturers to creating suitable software that gives the doctors the required information. At first, only sections of the data were considered, but with the increase in computing power, volume rendering has proved valuable. Also, the speedup from GPU-based implementations has improved the usability by allowing real-time viewpoint manipulations. However, we are convinced that, in certain situations, stereo is necessary for a good understanding.

Pre-operative planning, for example, aims at finding the optimal surgical procedure and path before the actual operation (e.g. in neurosurgery, orthopedic surgery and vascular surgery). The quantitative analysis of distances, volumes and angles is an essential part of this planning. Displaying the 3D volumes stereoscopically instead of monoscopically can substantially improve this analysis, especially if complex shapes and intertwined structures (e.g. blood vessels) are involved.

In medical education, a stereoscopic display can facilitate the understanding of complex spatial relationships or improve learning of skills needed for surgical procedures.

Stereoscopic visualization may increase the performance in diagnostic applications (e.g. decreased false positive and false negative diagnoses in cancer detection). However, since display quality is of highest importance for diagnosis, and since stereo display technology has by far not reached the same quality standards as 2D medical displays, stereoscopic diagnosis is limited to studies today. An overview of the use of stereoscopic displays in medical applications is given in [6].

As an example, consider Computed Tomography Angiography (CTA) where blood vessels are examined. Scans of the arteries in the brain and around the heart will typically result in very complex curving networks. The interactive volume rendering helps interpreting these structures, but it is still a demanding task. Take Figure 3(b) as an example: it is not obvious how the main arteries evolve. After adding stereo, like Figure 3(c), one can see[1] their curvatures more clearly.

B. Grain growth

The microstructure of many materials consists of grains with different crystallographic orientations. Under certain conditions, such as high temperature, the larger grains will grow at the expense of smaller grains. This phenomenon is called grain growth. A common technique to control the grain growth is the addition of impurities. This leads to the formation of second-phase particles. In [8], three-dimensional finite difference simulations are performed on a unit volume using periodical boundary conditions. Inspection of these simulations will improve the understanding of the interaction between the particles and the movement of the grain boundaries.

Original visualization in MATLAB resulted in images similar to Figure 4(a): only the structure at the system boundaries is shown. This provides some degree of information but definitely not enough to understand everything. Due to the slow three-dimensional graphics in MATLAB, no effort was made to improve this with more advanced visualizations.

After developing a suitable reader for the data files and designing an appropriate color scheme, the GPU accelerated volume rendering from our framework could be reused. This instantly resulted in an (almost) fifty-fold speedup of the visualization. But our framework has some more features. Composing for instance, introduces a totally new view on the problem, without any additional development cost. By using fully transparent colors for the grains, the boundaries (and thus parts of the inner structure) can be revealed (see Figure 4(b)).

It is obvious that rendering the edges semi-transparently would lead to incomprehensible images. Therefore, a circular shift is implemented along all three dimensions. This is accomplished by using an offset in the texture memory lookup rather then shifting the actual data. It allows users to browse through the volume. This way, the region of interest can be positioned at the best place in the view. This is important when special behavior at specific locations need to be illustrated.

As with previous examples, the exact location of the second phase particles is hard to determine from Figure 4(b). Because of the GPU acceleration, motion parallax becomes useful again and also occlusion can help to find their relative position of the impurities. However, adding stereo enables the interpretation of the absolute configuration. With a grayscale version of the colormap Figure 4(c) was created. It allows a more natural inspection of the particle-boundary intersection and facilitates the interpretation of their relative positions and orientations.

IV. Conclusion

As argued before and illustrated with two examples, stereo vision is a valuable addition to volume rendering. Although it might take a few hours to get used to it, the amount of information that can be transferred with these stereo images is much higher than with regular – mono – volume renderings.

Future work will focus on the quantification of the stereo effect and head tracking. The former focusses on assessing the quality and the level of information in the images; whilst the latter will enable the user to move his head to look 'around' the subject. This is a more intuitive way to interact with the viewpoint compared to rotating the subject using a regular input device. It should improve the interpretation even further.

Acknowledgment

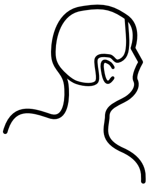
The authors are grateful to Dirk Loeckx of the Medical Imaging Center for the volumetric data and also like to thank Liesbeth Vanherpe for the work that led to the second example.

This work is supported by IWT (Flemish program: Institute for the Promotion of Innovation by Science and Technology in Flanders) in the context of the MEVIC project (Medical Virtual Imaging Chain).

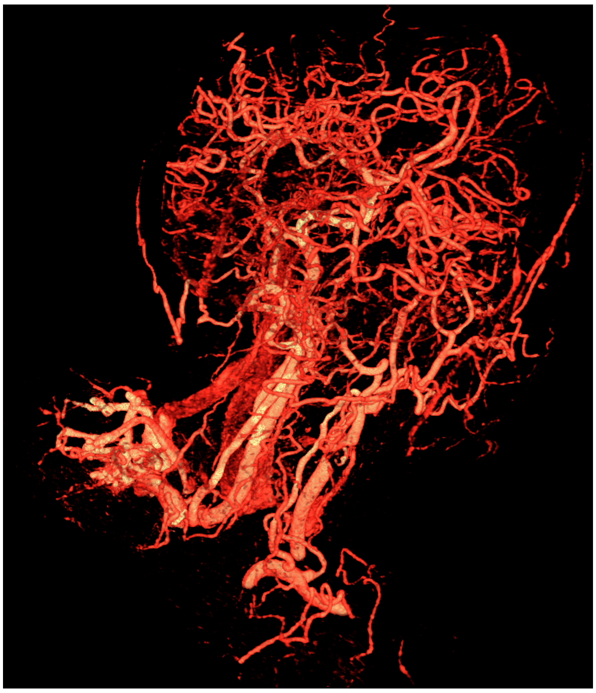
References

[1] P. Bourke. Creating and viewing anaglyphs, 2000.
[2] Klaus Engel, Markus Hadwiger, Joe M. Kniss, Aaron E. Lefohn, Christof Rezk Salama, and Daniel Weiskopf. Real-time volume graphics. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, page 29, New York, NY, USA, 2004. ACM.
[3] Farrel and Booth. *The Boeing Design Handbook for Imagery Interpretation Equipement*. Boeing Aerospace Co., Seattle, WA., 1984.
[4] Markus Hadwiger, Andrea Kratz, Christian Sigg, and Katja Bühler. Gpu-accelerated deep shadow maps for direct volume rendering. In *GH '06: Proceedings of the 21st ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, pages 49–52, New York, NY, USA, 2006. ACM.
[5] J. Kruger and R. Westermann. Acceleration techniques for gpu-based volume rendering. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 38, Washington, DC, USA, 2003. IEEE Computer Society.
[6] Haralambos Hatzakis Wijnand A. IJsselsteijn Maurice H.P.H. van Beurden, Gert Van Hoey. Stereoscopic displays in medical domains: A review of perception and performance effects. *Proceedings of the SPIE*, 7240:72400A–1 – 72400A–15, 2009.
[7] S. Stegmaier, M. Strengert, T. Klein, and T. Ertl. A simple and flexible volume rendering framework for graphics-hardware-based raycasting. *International Workshop on Volume Graphics*, 0:187–241, 2005.
[8] Liesbeth Vanherpe, Nele Moelans, Bart Blanpain, and Stefan Vandewalle. Bounding box algorithm for three-dimensional phase-field simulations of microstructural evolution in polycrystalline materials. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 76(5):056702, 2007.
[9] Ivan Viola, Armin Kanitsar, and Meister Eduard Gröller. Gpu-based frequency domain volume rendering. In *SCCG '04: Proceedings of the 20th spring conference on Computer graphics*, pages 55–64, New York, NY, USA, 2004. ACM.
[10] Colin Ware and Glenn Franck. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Trans. Graph.*, 15(2):121–140, 1996.
[11] P. Winner. Anaglyph methods comparison.
[12] Andrew Woods, Tom Docherty, and Rolf Koch. Image distortion in stereoscopic video systems. In *Stereoscopic Displays and Applications IV*, volume 1915, pages 36–48. SPIE- The International Society for Optical Engineering, 1993.

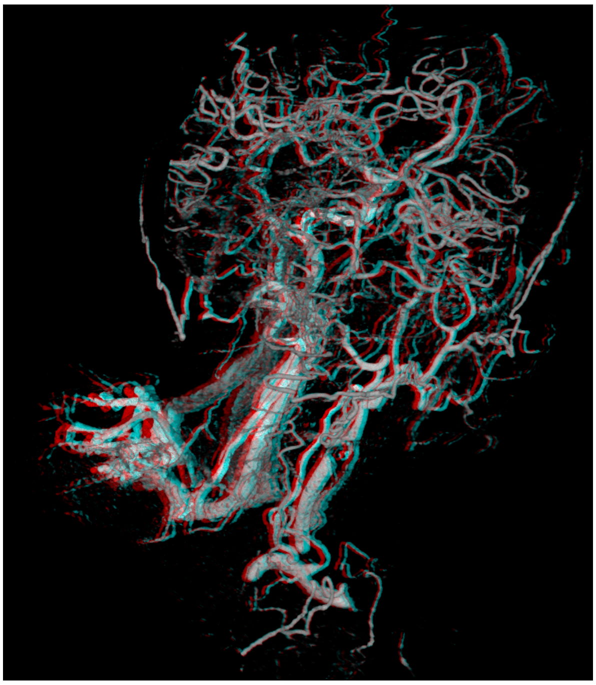
[1]Note that the stereo images here are only shown as anaglyph. Other techniques are not (yet) compatible with regular printing paradigms.



(a) Positioning the scan.

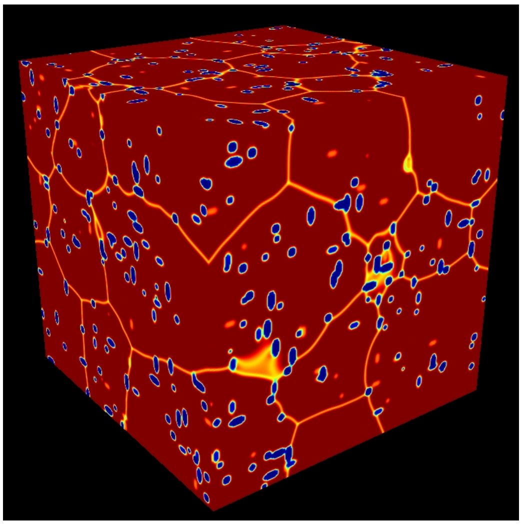


(b) Mono

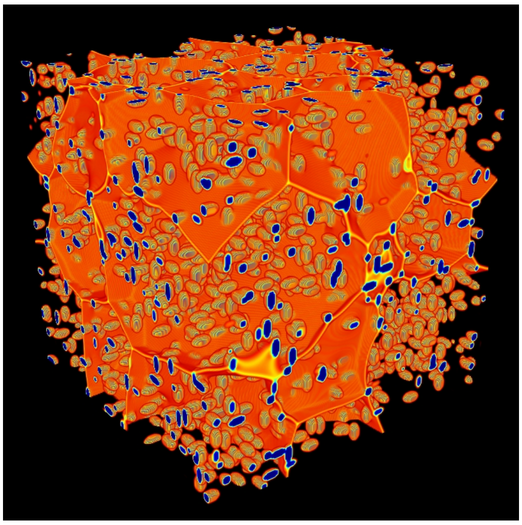


(c) Stereo anaglyph (left=red, right=cyan) and the *toe-in* projection. The vertical parallax can be observed on the left shoulder. The results for the asymmetric frustum and parallel axes projection mode are very similar.

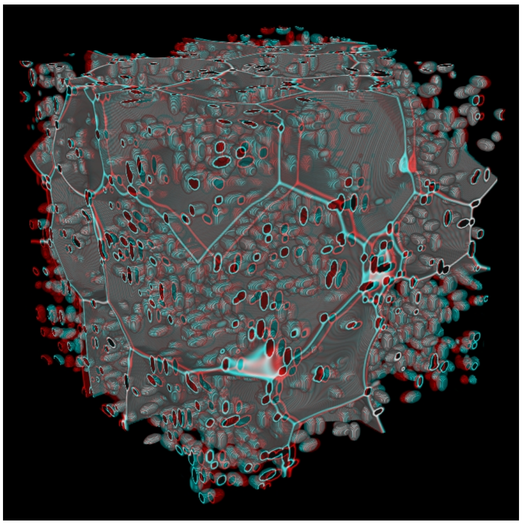
Figure 3. Illustration of stereo for Computed Tomography Angiography. The camera is positioned above the right shoulder, as shown on (a). (Data courtesy of the Medical Imaging Research Center, K.U.Leuven / UZ Gasthuisberg)



(a) Visualisation similar to MATLAB's results (red: grain, orange: boundaries, blue: second phase particles).



(b) Mono volume rendering with fully transparent grains. Note how parts of the inner structure have become visible.



(c) Stereo volume rendering using anaglyphs (left=red, right=cyan), the asymmetric frustum and parallel axes projection mode.

Figure 4. Illustration of volume rendering and stereopsis for grain growth. (Data courtesy of Liesbeth Vanherpe, K.U.Leuven, dept. Computer science)